

PROCEEDINGS OF SPIE

[SPIDigitalLibrary.org/conference-proceedings-of-spie](https://spiedigitallibrary.org/conference-proceedings-of-spie)

Stochastic optimization framework (SOF) for computer-optimized design, engineering, and performance of multi-dimensional systems and processes

Wolfgang Fink

Wolfgang Fink, "Stochastic optimization framework (SOF) for computer-optimized design, engineering, and performance of multi-dimensional systems and processes," Proc. SPIE 6960, Space Exploration Technologies, 69600N (15 April 2008); doi: 10.1117/12.784440

SPIE.

Event: SPIE Defense and Security Symposium, 2008, Orlando, Florida, United States

Stochastic Optimization Framework (SOF) for Computer-Optimized Design, Engineering, and Performance of Multi-Dimensional Systems and Processes

Wolfgang Fink^{a*}

^aCalifornia Institute of Technology, *Visual and Autonomous Exploration Systems Research Laboratory*, Division of Physics, Mathematics & Astronomy, 1200 E California Blvd,
Mail Code 103-33, Pasadena, CA 91125, USA

ABSTRACT

Many systems and processes, both natural and artificial, may be described by parameter-driven mathematical and physical models. We introduce a generally applicable Stochastic Optimization Framework (SOF) that can be interfaced to or wrapped around such models to optimize model outcomes by effectively “inverting” them. The *Visual and Autonomous Exploration Systems Research Laboratory* (<http://autonomy.caltech.edu>) at the California Institute of Technology (Caltech) has long-term experience in the optimization of multi-dimensional systems and processes. Several examples of successful application of a SOF are reviewed and presented, including biochemistry, robotics, device performance, mission design, parameter retrieval, and fractal landscape optimization. Applications of a SOF are manifold, such as in science, engineering, industry, defense & security, and reconnaissance/exploration.

Keywords: Multi-parameter optimization, design/performance optimization, gradient-based steepest-descent methods, local minima, global minimum, degeneracy, overlap parameter distribution, fitness function, stochastic optimization framework, Simulated Annealing, Genetic Algorithms, Evolutionary Algorithms, Genetic Programming, Evolutionary Computation, multi-objective optimization, Pareto-optimal front, trade studies

1. INTRODUCTION

Many systems and processes, both natural and artificial, may be described by parameter-driven mathematical and physical models. These models are usually exercised in a forward-fashion: A certain set of values for the model-intrinsic parameters yields a corresponding outcome when processed through the model at hand. It is usually a straightforward task to define what an optimal outcome of the model is. Conversely, it is incomparably more difficult, if not impossible in many cases, to determine what the corresponding parameter values are that, when applied to the model, yield this outcome, or approximate it as closely as possible. In addition, if more than one set of parameter values yields the same desired outcome, the model is degenerate. One way to determine the optimal parameter values is to analytically invert the models, or to run them backwards. In many cases this is analytically or practically infeasible due to the model-related complexity and high degree of non-linearity. To overcome this inherent problem, we introduce a generally applicable *Stochastic Optimization Framework (SOF)* that can be interfaced to or wrapped around such models to effectively “invert” them.

2. STOCHASTIC OPTIMIZATION FRAMEWORK (SOF)

A Stochastic Optimization Framework (SOF, Fig. 1) allows for efficient sampling of the entire model-intrinsic parameter space by repeatedly running the respective model forward (e.g., on a single, cluster, or parallel computer) and by comparing the outcomes against the desired outcome, which results in a fitness measure. The goal of the SOF is to optimize this fitness. This approach is in sharp contrast to optimizing around a point design, which is often the case in engineering. Deterministic optimization techniques, such as gradient-based steepest-descent methods, are powerful and

* e-mail: wfink@autonomy.caltech.edu; phone: +1-626-395-4587; website: <http://autonomy.caltech.edu>

efficient in problems that exhibit only few local minima in the solution space. However, when dealing with multiple or infinite numbers of local minima, heuristic stochastic optimization methods, such as Simulated Annealing [1, 2] related algorithms, Genetic Algorithms [3, 4], other Evolutionary Algorithms, and Genetic Programming [5], may become the prime methods of choice because of their capability to overcome local minima. In our case we choose modified Simulated Annealing algorithms as the optimization engine for SOFs.

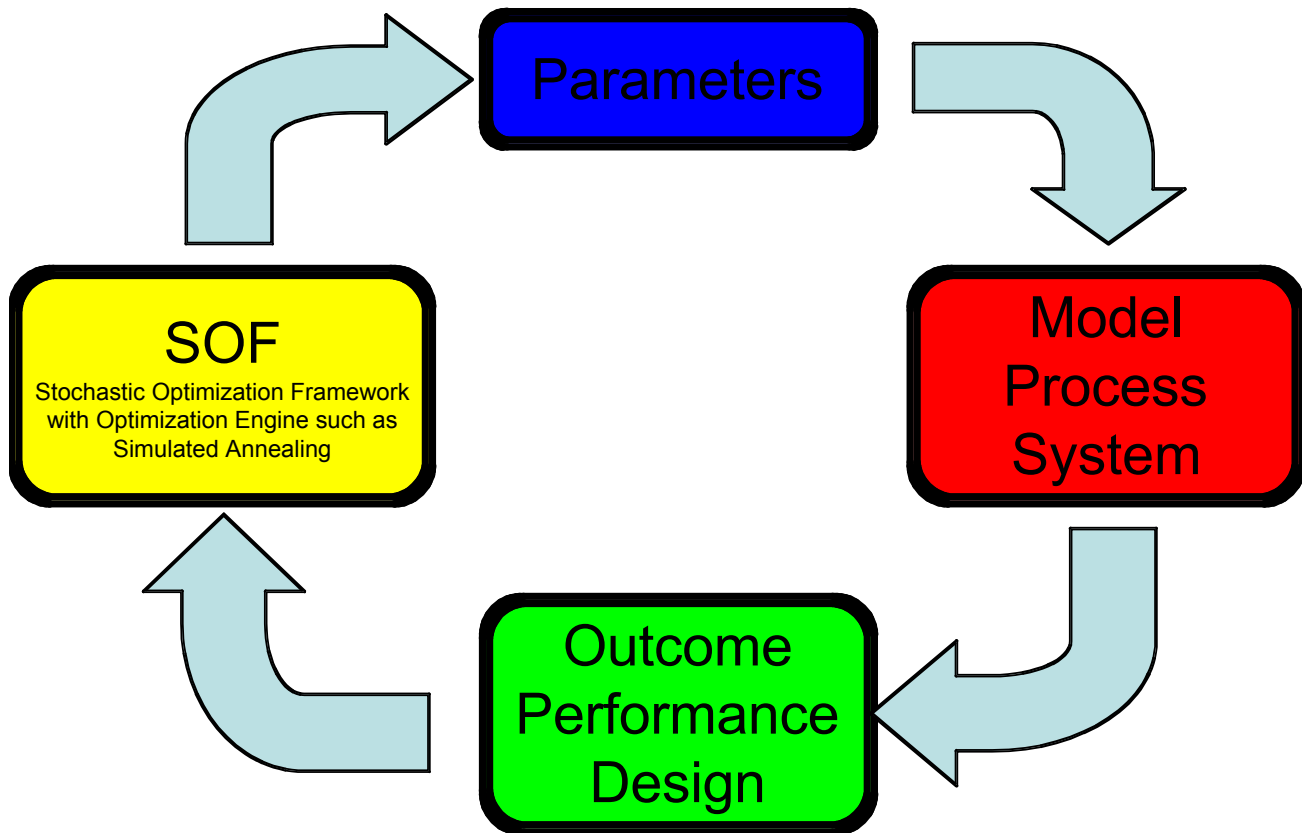


Fig. 1. Functional schematic of a Stochastic Optimization Framework (SOF): The SOF efficiently samples the entire model/process/system-intrinsic parameter space by repeatedly running the respective model/process/system forward (e.g., on a single, cluster, or parallel computer) and by comparing the outcomes against a desired outcome, which results in a fitness measure. The goal of the SOF is to optimize this fitness by using (in our case) modified Simulated Annealing algorithms as the optimization engine.

2.1 Optimization Engine – Simulated Annealing (SA)

Simulated Annealing (SA) [1, 2] is a widely used and well-established optimization technique, especially for high-dimensional configuration spaces. The goal is to minimize an energy/fitness function E , which is a function of N variables or parameters. The iterative minimization process is performed by randomly changing the values of one or more of the N variables within their respective, defined value ranges, and by subsequently reevaluating the energy function E per iteration step. Two cases can occur:

- (1) The change in the variable values results in a new, lower energy function value;
- (2) The energy function value is higher or unchanged.

In the first scenario the new set of variable values is stored and the change accepted. In the second scenario, the new set of variable values is only stored with a certain likelihood (Boltzmann probability, defined by an annealing temperature). This ensures that the overall optimization process does not get trapped in local minima too easily such as is the case

with, for example, gradient-based, steepest-descent (“greedy”) downhill optimization (Fig. 2). The annealing temperature directly influences the Boltzmann probability by making it less likely to accept an energetically unfavorable step, the longer the optimization lasts (also known as the “cooling schedule”). Then the overall procedure is repeated until the annealing temperature has reached its end value, or a preset number of iterations has been exceeded, or the energy function E has reached an acceptable user-defined level.

Furthermore, Simulated Annealing, as opposed to Genetic Algorithms and other population-based Evolutionary Algorithms, is characterized by very few user-defined parameters, mostly pertaining to the cooling-schedule, making it potentially more straightforward to tackle the actual optimization problem rather than the intricacies of the optimization algorithm used.

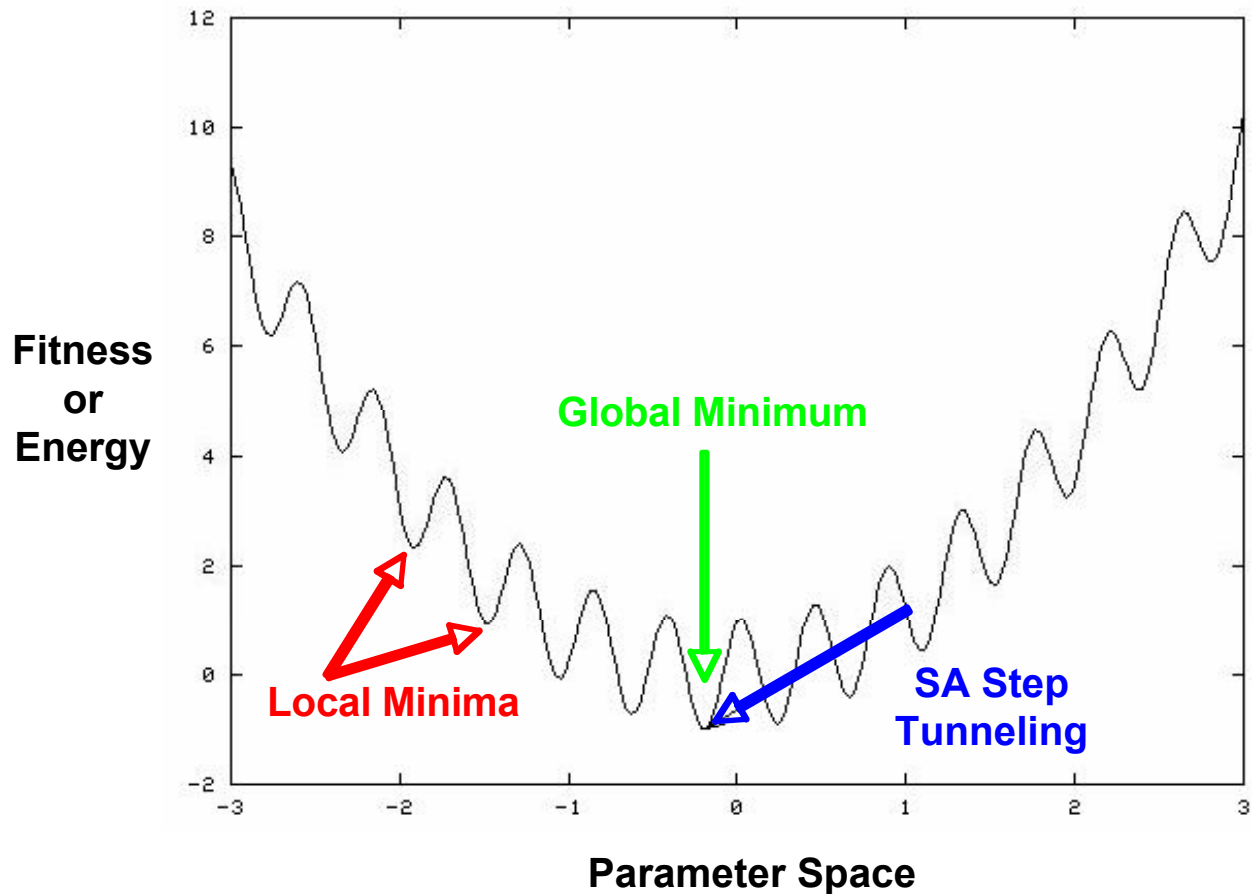


Fig. 2. Example of a fitness or energy landscape with multiple local minima. Deterministic, gradient-based, steepest-descent (“greedy”) downhill optimization algorithms tend to get trapped in local minima without ever reaching the global minimum. Heuristic, stochastic optimization algorithms, such as Simulated Annealing [1, 2], Genetic Algorithms [3, 4], and other Evolutionary Algorithms possess mechanisms to “tunnel” through fitness/energy barriers to reach the global minimum, or at least to not get trapped in local minima for too long.

2.2 Parallelization Efficiency of SA

A major advantage of using Simulated Annealing as the optimization engine at the core of a SOF lies in the fact that it is “embarrassingly” parallel in nature: As many independent SA runs can be started as there are CPUs available without any message passing as opposed to population based Genetic Algorithms or other Evolutionary Algorithms that require, at the very least, the passing of fitness evaluation information. As a result, an almost perfect linear speed up can be

expected (with the exception of start-up time) since each processor performs an independent optimization run without cross-communication (e.g., message passing) between processors (Fig. 3). We have demonstrated the parallel deployment of Simulated Annealing on 1,002 CPUs of the institutional 1,024 CPU cluster computer COSMOS at NASA's Jet Propulsion Laboratory (see also Section 3.4).

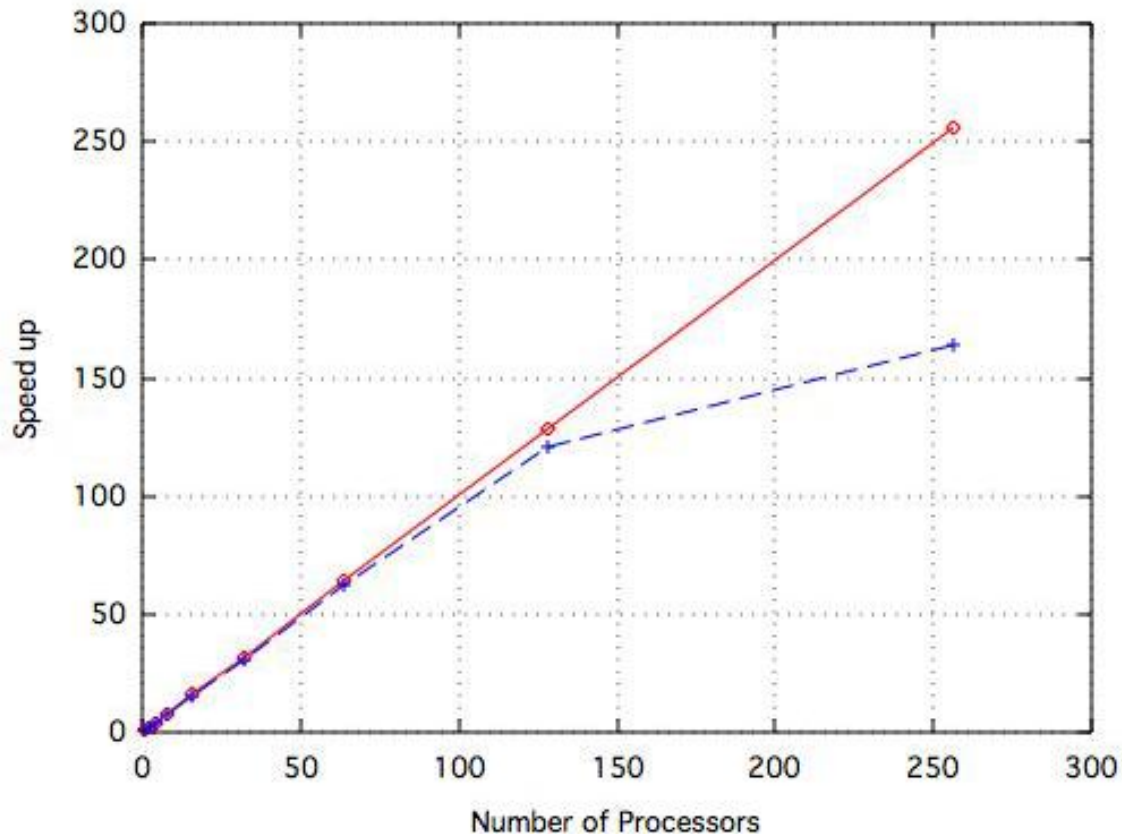


Fig. 3. Computational speed up with “embarrassingly” parallel Simulated Annealing runs for low-thrust trajectory optimization (from [6], see also Section 3.4): Each processor was given its own, independent Simulated Annealing run (with a different random number seed, respectively) without any cross-communication (e.g., message passing). As can be seen, a “perfect” linear speed up was obtained for up to 128 processors, which became less efficient (64%) when utilizing 256 processors (the red solid line is the theoretical perfect linear speed up, and the blue dashed line is the experimental result). The predominant reason for the latter behavior in this particular SA run appeared to have been the initial start-up time required to distribute and start the individual jobs on all 256 processors from a single node.

2.3 Characterization of Degeneracy in Solution Spaces – Overlap Parameter Distribution (OPD)

As mentioned above, if more than one set of parameter values yields the same desired outcome in a model/process/system, the model/process/system is degenerate. To characterize the degeneracy one may apply methods borrowed from Statistical Mechanics for the description of multi-body systems, such as the *Overlap Parameter Distribution (OPD)*. OPD is a well-established technique for characterizing the similarity of solution vectors in high-dimensional parameter spaces, most prominently for the characterization of the solution/coupling space for artificial neural networks (see [7, 8]). The OPD is determined by calculating the *overlap parameters*, i.e., the N-dimensional scalar products of appropriately normalized solution vectors, for all possible pair-wise combinations of the solutions produced through a SOF. The resulting OPD may be indicative of the existence or the degree of degeneracy among the solutions, expressed as the number of centers in the distribution (i.e., multi-center distribution) and/or the distribution width (see also Section 3.5). This is independent of any assumptions.

2.4 Multi-objective Optimization and Trade Studies – Pareto-Optimal Front (POF) of Non-dominated Solutions

The concept of *Pareto-optimal fronts* (POF) is a useful tool when dealing with multi-objective optimization scenarios, i.e., scenarios where multiple, often conflicting or mutually exclusive objectives have to be satisfied/optimized. Let us consider the example of low-thrust trajectory optimization (see also Section 3.4). The calculation/design and optimization of low-thrust trajectories is called for when dealing with low-thrust elements, i.e., ion-engines (e.g., NASA's Deep Space 1 and ESA's SMART-1 missions) and when the amount of remaining propellant at the target location (e.g., target orbit) is to be maximized. The general goal is to maneuver a spacecraft with a series of thrust arcs from position A to position B in the most fuel-efficient and simultaneously time-efficient manner. Clearly the optimum exists somewhere in between the following two extreme cases: (a) thrust continuously and thereby reach the destination in the shortest amount of time at the expense of maximum fuel usage, or (b) fly infinitely long and use almost no fuel by exploiting the gravitational pull of planetary bodies (e.g., swing-by maneuvers). As can be seen, the fuel consumption and the flight time are two objectives that are conflicting, because a 0 fuel consumption and 0 flight time mission obviously does not exist. Hence, there is a tradeoff between these two objectives and as a result an entire front – the so-called Pareto-optimal front (here: curve in the two-dimensional space of flight time and fuel consumption) – of possible mission scenarios unfolds: a feature that allows for *trade studies*. The mission scenarios (i.e., points along the Pareto-optimal front) are *non-dominated* in terms of (fuel consumption, flight time)-combinations: for any given flight time no other solution (here: low-thrust trajectory) exists that would have a lower fuel consumption and vice versa. These solutions are said to be Pareto-optimal. The same concept of a Pareto-optimal front, shown here for two objectives, can be generalized to higher numbers of objectives, i.e., multi-objective optimization in multi-dimensional parameter spaces. It should be noted that Pareto-optimal fronts do not necessarily have to be continuous or monotonic.

While Genetic Algorithms, Evolutionary Algorithms, and other population-based algorithms, per definitionem, are directly capable of generating Pareto-optimal fronts, this capability can be attained as well with Simulated Annealing algorithms, by exploiting the parallelization efficiency of SA as outlined in Section 2.2.

3. SOF APPLICATION EXAMPLES

The *Visual and Autonomous Exploration Systems Research Laboratory* (<http://autonomy.caltech.edu>) at the California Institute of Technology (Caltech) has long-term experience in the optimization of multi-dimensional systems and processes. In the following, we briefly review and present several diverse application examples, the research for which was carried out at Caltech or at NASA's Jet Propulsion Laboratory (JPL) at the time, where we successfully employed Stochastic Optimization Frameworks in conjunction with Simulated Annealing, including biochemistry, robotics, device performance, mission design, parameter retrieval, and fractal landscape optimization.

3.1 SOF Application Example #1: Biochemistry – De Novo Protein Design & Effective Sequence Search

In nature, proteins become biologically active once they are folded into a certain three-dimensional structure, i.e., tertiary structure. The discipline of *protein folding* tries to predict the tertiary structure of a given sequence of amino acids from first principles. This is a vastly complex and computationally demanding undertaking. *De novo protein design* [9], on the other hand, consists of two steps: (1) selecting a desired tertiary structure (i.e., representative of a desired biological activity) and (2) finding a sequence of so-called rotamers (i.e., amino acid side-chains and their respective positions) that would stabilize this three-dimensional fold. Therefore, the task is to find the global energy minimum of a given protein by choosing the appropriate rotamers at each residue (i.e., protein site) with respect to their self-energy and interaction energy that are given by theoretical chemical models (e.g., models using Lennard-Jones 6-12 interaction potentials). This approach relies on the assumption that the global energy minimum corresponds to the most stable fold of the protein, which can subsequently be synthesized and tested experimentally in the laboratory. Due to the vast number of possible rotamer combinations, exhaustive search and other deterministic algorithms, such as dead-end elimination (DEE) [10], for exploring all possibilities are naturally limited to relatively small protein sizes. For biologically interesting proteins (i.e., sizes larger than 50 residues) one may have to rely increasingly on non-deterministic methods, e.g., mean-field approach, Monte Carlo methods, Simulated Annealing, Genetic Algorithms, and related algorithms [11]. At Caltech, we have developed a modified Simulated Annealing algorithm as part of a SOF that is capable of tackling larger proteins (50 residues and more) by identifying the appropriate rotamers at each residue of the protein backbone [12]. For a 50

residues protein with an average of 100 rotamers per residue the complexity (i.e., total number of possible rotamer combinations across the protein) exceeds already 10^{100} . This capability may be beneficial for biochemistry, biomedicine, and the pharmaceutical industry.

3.2 SOF Application Example #2: Robotics – Robotic Limb Deployment

Extra-terrestrial space exploration has created the need for robotic reconnaissance operations in extreme environments such as planetary atmospheres, surfaces, and subsurfaces. Robotic reconnaissance is needed as well in potentially hazardous or inaccessible operational areas on Earth, such as those related to military or terrorist activities, or areas that have been exposed to biochemical agents, radiation, or natural disasters. Furthermore, robotic limb deployments are abundant in industry (e.g., automobile industry), medicine (e.g., surgical robots), and other civilian operations.

From a space-exploration perspective current and future planetary surface and subsurface robotic exploration missions (e.g., [13-16]) involving planetary landers, such as the NASA Phoenix mission currently en route to Mars, and/or rover-type vehicles such as Mars Exploration Rover (MER), Mars Science Laboratory (MSL), and subsurface access vehicles are and will be equipped with robotic arms with four or more joints (DOF), each joint having a number of discrete angle positions. Fast and efficient safe rover movement (e.g., legged and cliff-scaling rovers [17]) and rover or lander arm deployment algorithms, taking rover position and surrounding ground obstacles (e.g., rocks) into account, that can be executed with onboard CPU power will tremendously enhance mission autonomy by reducing the number of up- and downlink events, and thus increase the useful lifespan of a mission.

The calculation of a collision-free robotic limb deployment path is a search in a high-dimensional configuration space that, in addition, may be constrained by self-collisions and/or collisions with surrounding obstacles (e.g., rocks on a planetary surface). A robotic limb comprising N joints or degrees-of-freedom (DOF) with, e.g., 100 discrete stepper motor angle positions per joint, already spans a configuration space of $10^{(2*N)}$. With $N > 6$, the number of possible configurations lies already beyond exhaustive search in a timely manner, especially on computationally constraint platforms. To increase the degree of complexity even more, the deployment of a robotic limb usually requires the generation/calculation of a series of valid and safe configurations in order to move the limb from a start position/configuration to an end position/configuration, i.e., the “safe” robotic limb deployment path, whereby “safe” is defined as both mechanically feasible and collision-free.

Using a Stochastic Optimization Framework (SOF) for space systems and space applications [18-20], we have developed an efficient, modified Simulated Annealing algorithm (details provided in [18]), normally used in computationally highly expensive optimization and search problems, such as Traveling Salesman Problem and De Novo Protein Design ([12], see also Section 3.1), for operating complex mobile systems with several degrees of freedom, such as robotic limbs.

In particular our SA algorithm, applied to a planetary rover equipped with an instrument-bearing robotic arm, performs the following tasks [18-20]: (1) Calculation of a safe robotic arm end configuration in joint angle space for science instrument placement with respect to a science target of interest (e.g., rock on planetary surface), given by a predetermined x-y-z end position of the instrument-carrying joint together with a surface normal at that point for instrument orientation with respect to the target; (2) calculation of a safe deployment path (i.e., collision-free and mechanically feasible set of robotic arm deployment segments) exclusively in joint angle space from a start robotic arm configuration into the pre-calculated end configuration determined in (1); (3) optimization of deployment paths, using step (2) iteratively together with a clean-up procedure [18], with respect to minimizing the absolute sum of overall joint angle movement, i.e., shortest/fastest deployment path; and (4) optimization of deployment paths, using step (2) iteratively together with a clean-up procedure [18], with respect to minimizing the joint angle movement of the individual joints between deployment steps for reducing wear and tear – an important aspect for remote planetary operation to prolong the mission lifetime, increase science return, and thereby increase overall mission cost efficiency.

The SOF-based robotic limb deployment optimization approach [18] offers an alternative to conventional, time-intensive, deterministic algorithmic approaches, such as robotic arm deployment path planning algorithms based on *inverse kinematics* [21]. Moreover, the SA algorithm employed [18] is suitable to be run onboard industrial robots, robots in telemedicine, remote spacecraft, planetary landers, or rovers, i.e., computational low-power computing platforms. The SA-based rover arm path planning algorithm has been successfully tested onboard a rover platform [22, 23] and with a software simulator at JPL.

3.3 SOF Application Example #3: Device Performance – Fully Automated Tuning of MEMS Micro-Gyroscopes

Since its advent in the late 1980s, MEMS development has progressed by leaps and bounds and now encompasses a diverse range of sensors and actuators [24]. MEMS devices, due to their inherently low mass, size, and power, are ideal for both space applications as well as sensor network applications. A number of MEMS devices have been specifically developed for the purpose of robotic planetary exploration [25, 26]. MEMS technologies have been developed for a diverse set of applications. For space exploration and defense & security purposes, MEMS sensors can be categorized under two classes: physical sensors and chemical sensors. The early successes in MEMS technology were in the area of physical sensors. As an example, we focus in the following on a physical sensor for sensing rotation: a MEMS micro-gyroscope [27, 25].

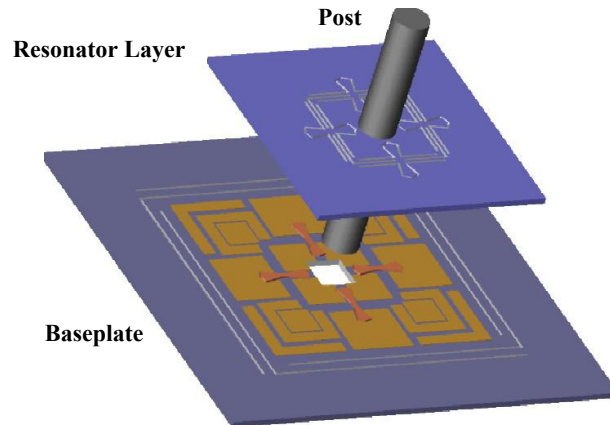


Fig. 4. Schematic, exploded view of a single-axis gyroscope (from [28]) developed at NASA's Jet Propulsion Laboratory (JPL) [25]. The inertial sensing element is the central post, about whose axis the rotation is sensed. The post is mounted on a layer containing in-plane orthogonal resonators. The post/resonator assembly is suspended over a substrate containing an arrangement of multiple electrodes for actuation, sensing and tuning the frequencies of the resonance modes. The gyroscope operates by "rocking" the post about an in-plane axis and consequently sensing the deflection about the orthogonal in-plane axis generated by the Coriolis force. The novel design is aimed both at closely matching the orthogonal resonance frequencies, and at maximizing the quality factor by minimizing mechanical coupling losses to the frame. The ultimate performance of the gyroscope has a strong dependence on both of these factors above.

The MEMS post-resonating micro-gyroscope (Fig. 4), developed by the MEMS Technology Group at JPL [25], contains a central post as the inertial sensing element, about whose axis the rotation is sensed. The post is mounted on a layer containing in-plane orthogonal resonators. The post/resonator assembly is suspended over a substrate containing an arrangement of multiple electrodes for actuation, sensing and tuning the frequencies of the resonance modes. The gyroscope operates by "rocking" the post about an in-plane axis and consequently sensing the deflection about the orthogonal in-plane axis generated by the Coriolis force. The novel design is aimed both at closely matching the orthogonal resonance frequencies, and at maximizing the quality factor by minimizing mechanical coupling losses to the frame. The ultimate performance of the micro-gyroscope has a strong dependence on both of these factors above and it is therefore subject to an electro-static fine-tuning procedure, performed by hand, which is necessary due to unavoidable manufacturing inaccuracies. To fine-tune the micro-gyroscope, four voltages applied to eight electrodes (capacitor plates) have to be determined within a range of -60V to 15V , respectively [29-33]. The fine-tuning directly correlates with the accuracy of the micro-gyroscope in later use.

The process of manually fine-tuning the micro-gyroscope takes on the order of several hours. To fully automate this time-taking process, we have established a hardware/software interface (as part of the SOF) to the existing manual micro-gyroscope tuning hardware-setup, using commercial-off-the-shelf (COTS) components such as four programmable power supplies, one offset power supply, and an (electronic) signal analyzer, as well as driver and analyzing software [29-33].

We have developed a modified Simulated Annealing algorithm for efficiently determining optimized tuning voltages [29-33] and incorporated it in the hardware/software interface of the SOF. We subsequently were able to successfully fine-tune, for the first time fully automatically, both MEMS post-resonating micro-gyroscopes and MEMS disc-resonating micro-gyroscopes within one hour to a level of accuracy that is equal to or better than what can be accomplished manually [29-33] (Fig. 5).

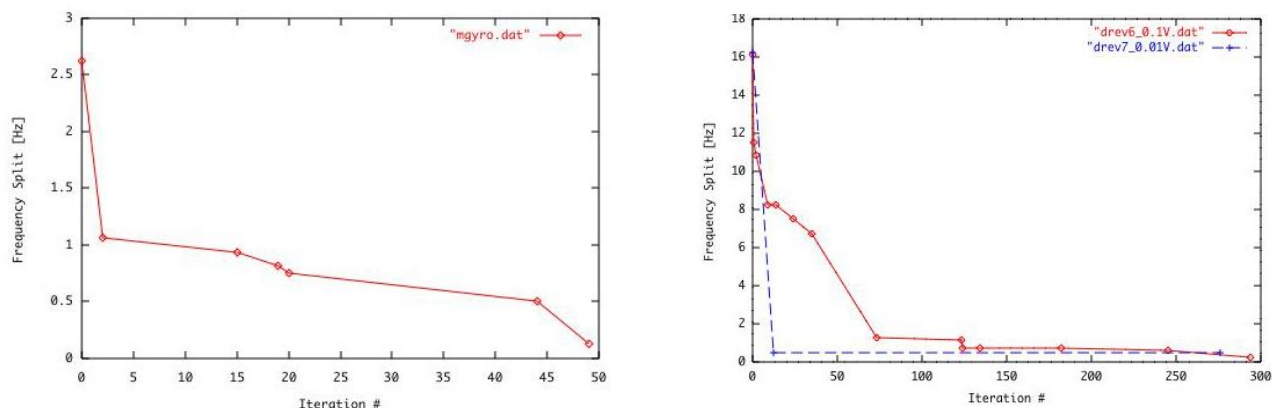


Fig. 5. Frequency split as a function of Simulated Annealing iterations within a Stochastic Optimization Framework (from [29-33]) to fully automatically fine-tune a MEMS post-resonating micro-gyroscope (left) and a MEMS disk-resonating micro-gyroscope (right).

The novel capability of fully automated micro-gyroscope tuning enables ultra-low mass and ultra-low-power high-precision Inertial Measurement Unit (IMU) systems to calibrate themselves automatically, for example, during deployment aboard UAVs, missiles, and spacecraft.

3.4 SOF Application Example #4: Mission Design – Low-Thrust Trajectory Design and Optimization

As pointed out already in Section 2.4, the design and optimization of low-thrust trajectories is called for when dealing with low-thrust elements, i.e., ion-engines (e.g., NASA's Deep Space 1 and ESA's SMART-1 missions) and when the amount of remaining propellant at the target location (e.g., target orbit) is to be maximized. The general goal is to maneuver a spacecraft with a series of thrust arcs from position A to position B in the most fuel-efficient and simultaneously time-efficient manner. The task is then to determine the number, onset times, and durations of so-called thrust arcs along the trajectory leading from position A to target position B (e.g., target orbit). In between thrust arcs the spacecraft is coasting (i.e., moving without power along so-called coast arcs). In addition to determining the occurrence of thrust arcs, the spatial orientation of the thruster(s) for each thrust arc has to be specified as well. Ion-engine propulsion can be treated in a binary fashion: either the ion-engine is ON (i.e., at full thrust), or it is OFF (i.e., at no thrust). The resulting trajectory can be computed by integrating the Newtonian equations of motion over the determined set of thrust and coast arcs. The difference between the so-reached end position and the target position delivers the fitness measure necessary for implementing a SOF.

As such we have successfully developed and implemented, as part of a SOF, a modified Simulated Annealing algorithm for low-thrust trajectory optimization for orbit transfers [34, 35, 6]. Some of these optimizations [6] comprised the parallel deployment of SA on the institutional 1,024 CPU cluster computer COSMOS at JPL (see also Fig. 3). Furthermore, Pareto-optimal fronts of possible mission scenarios were computed [35, 6] that allowed for mission trade studies (see also Section 2.4).

3.5 SOF Application Example #5: Parameter Retrieval – Spectral Retrieval

When measuring, for example, a planetary spectrum (e.g., Earth spectrum from orbit, earthshine spectrum from the ground, martian spectrum, or extra-solar planet spectrum) the goal is to determine, which atomic and molecular spectra, when superimposed in the correct ratio, would yield the observed spectrum. This process of *spectral retrieval* is essential for future missions such as NASA's Terrestrial Planet Finder (TPF). Such missions, in the hunt for extra-solar planets, will gather spectral information of potential candidates that will be averaged over the visible planetary disk and the exposure time of the observation. Because of the immense distances, this spectral information will most likely be confined to one imaged pixel. By means of radiative transfer models, scientists are able to make predictions as to what kind of spectrum to expect when parameters for a certain type of planet (i.e., planet with a certain surface composition such as soil, water, ice, vegetation, etc.) are entered into the model. With such parameters the radiative transfer models generate synthetic spectra that can be compared to the actual, observed spectra. Ideally one would want to invert such a radiative transfer model in order to retrieve the spectral parameters that would best explain/fit the observed spectrum, and hence would help characterize the planet's surface composition. However, this is most of the time analytically or practically infeasible due to the inherent complexity and high degree of non-linearity of radiative transfer models.

For this purpose, we have successfully developed and implemented a SOF for spectral retrieval [36-39], using a modified Simulated Annealing algorithm and other evolutionary computing techniques (e.g., genetic algorithms) for the automatic retrieval of spectral parameters (e.g., atmospheric and surface-related parameters such as cloud and surface types [39]) for the earthshine spectrum recorded by Woolf et al. (2002) [40]. Preliminary results demonstrate a least square fit accuracy of 10% between the observed spectrum and the synthetic spectra generated via the radiative transfer model used [39].

Furthermore, there is a potential for encountering degeneracy in the solution space of retrieved spectral parameters [39]: More than one planetary surface composition may yield an equally good fit of the observed spectrum (within defined accuracy limits) instead of just one unique planetary surface composition. This has a potentially huge impact on the feasibility and success of future missions that search for extra-solar planets (e.g., TPF). To investigate this degeneracy, the aforementioned Overlap Parameter Distribution (OPD) technique can be employed. An additional technique for the visualization of degeneracy is Principal Component Analysis (PCA) [39].

OPD, as opposed to PCA, is computationally inexpensive. If N is the number of spectral parameters, they can be mathematically expressed as an N -dimensional vector. Each degenerate solution to the spectral retrieval problem (i.e., fit of the observed spectrum via the radiative transfer model) will yield such a vector. The subsequent calculation of the N -dimensional scalar products of the appropriately normalized solution vectors, for all possible pair-wise combinations of the solutions produced through the SOF, will result in the OPD. If the resulting OPD exhibits only a single, sharply peaked distribution close to 1 (if the solution vectors are normalized to 1), the degeneracy among the solutions consists of only slight variations around a typical, average solution [41]: the planetary surface composition will be more or less unique. This would be a benign (almost ideal) scenario for missions searching for extra-solar planets. If, however, the resulting OPD exhibits more than one center (e.g., multi-center distribution), or is not sharply peaked, nor located close to 1, this would be indicative of true degenerate behavior [41]: more than one planetary surface composition would explain the observed spectrum, and these compositions could be qualitatively different, depending on the width of the OPD or the number of centers in the OPD. It is this latter case that would pose a significant problem for future missions searching for extra-solar planets, such as TPF, and would have to be addressed, e.g., by improving, modifying, or changing the radiative transfer model used.

3.6 SOF Application Example #6: Fractal Landscape Optimization – WCCI/CEC “Huygens Probe” Competition

The IEEE Congress on Evolutionary Computation (CEC) International “Huygens Probe” Optimization Competition, designed and organized by Cara MacNish [42], senior lecturer at the University of Western Australia School of Computer Science and Software Engineering, was first held at the 2006 IEEE World Congress of Computation Intelligence (WCCI) in Vancouver, Canada. The competition addressed a “worst case” scenario optimization problem of finding the lowest point possible in a fractal energy/fitness landscape within 1,000 iteration steps (Fig. 6). The energy/fitness landscape was a function of two variables x and y . The fractal property of the landscapes ensured that even as optimization algorithms narrowed their searches to smaller regions, the landscape appeared just as rugged and the optimization problem remained just as hard. In fact the scale invariance of the fractal landscapes continued to the limit of IEEE 64-bit floating-point resolution. Optimization algorithms that do well on these kinds of problems are

therefore expected to generalize more easily to problems with more limited scale. In the “Huygens Probe” optimization scenario the fitness landscape was never revealed. Only the depth z for a submitted (x, y) -location per iteration was reported back to the optimization algorithm. For the competition, access was given to a set of 20 unseen fractal surfaces. All entries were ranked according to their performance on each of the 20 surfaces, and the algorithm with the highest overall average ranking was the winner. We applied a modified Simulated Annealing algorithm that allowed for both global and local search at an increasingly more refined scale. This algorithm won the “Huygens Probe” Optimization Competition in 2006 with 11 out of 20 direct wins (1st place) and an average ranking of 2.4 [43].

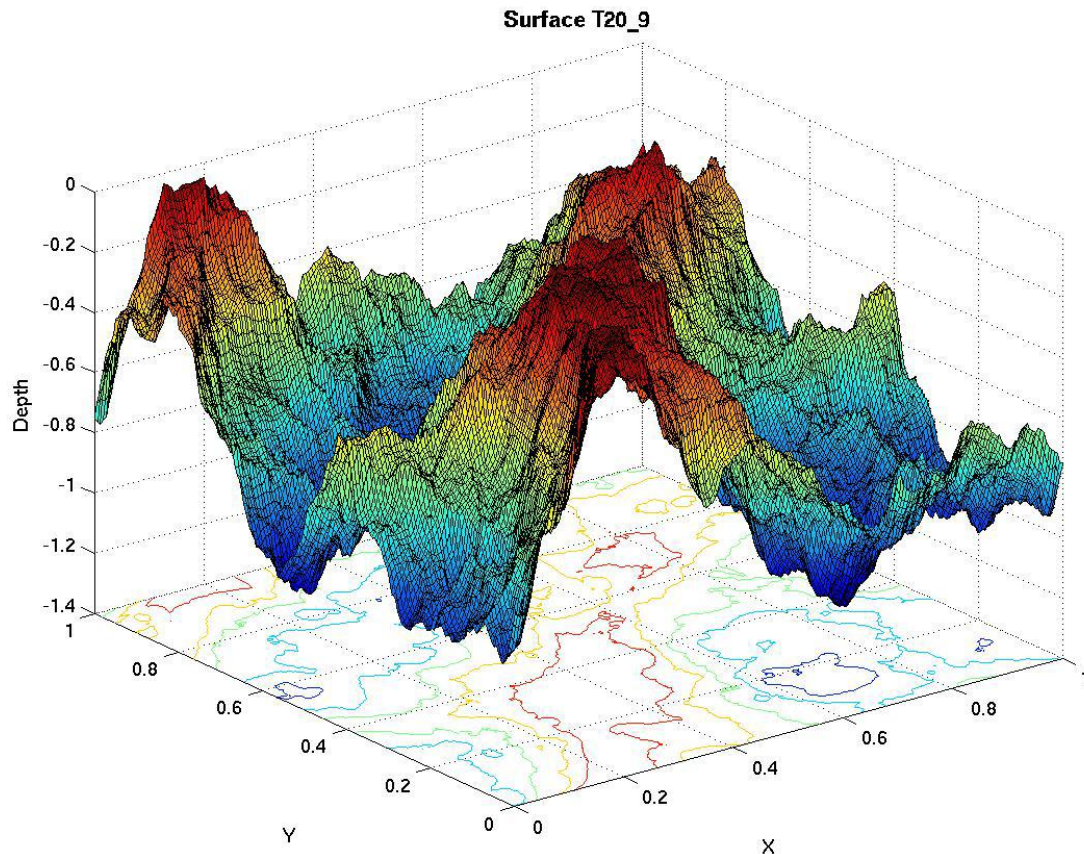


Fig. 6. Example self-similar (fractal) island from the IEEE Congress on Evolutionary Computation (CEC) International “Huygens Probe” Optimization Competition [42], held at the 2006 IEEE World Congress of Computational Intelligence (WCCI) in Vancouver, Canada. [Image courtesy Cara MacNish, University of Western Australia]

4. DISCUSSION & OUTLOOK

Stochastic Optimization Frameworks (SOF) in conjunction with efficient optimization engines, such as Simulated Annealing, are powerful in that they can routinely match or exceed human expert performance in the optimization of processes, system design and performance. For example: The fully automated tuning of MEMS micro-gyroscopes, the path planning for the deployment of robotic arms, and the optimization of low-thrust trajectories contributed to co-winning the “Humie Silver Award” for demonstrating Human Competitive Performance in the Automated Design of Space Systems from the Genetic and Evolutionary Computation Conference (GECCO) 2005. As seen from the examples presented above, the applications of SOFs are manifold, such as in science, engineering, industry, defense & security, and reconnaissance/exploration.

5. ACKNOWLEDGEMENTS

Some of the SOF application examples reviewed and described in this publication were originally carried out by us at the time at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration and supported by the JPL Research and Technology Development Program.

REFERENCES

1. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of State Calculation by Fast Computing Machines, *J. of Chem. Phys.*, 21, 1087 – 1091, 1953.
2. S. Kirkpatrick, C.D. Gelat, M.P. Vecchi, Optimization by Simulated Annealing, *Science*, 220, 671 – 680, 1983.
3. J.H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan, 1975.
4. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
5. J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: The MIT Press, 1992.
6. S. Lee, R.P. Russell, W. Fink, R.J. Terrile, A.E. Petropoulos, P. von Allmen, Low-Thrust Mission Trade Studies with Parallel, Evolutionary Computing, *IEEE Aerospace Conference Proceedings*, Big Sky, Montana, 2006.
7. J. Hertz, A. Krogh, R.G. Palmer, *Introduction To The Theory Of Neural Computation*, Lecture Notes Volume I, Addison-Wesley Publishing Company, 1991.
8. B. Müller, J. Reinhardt, *Neural Networks: An Introduction*, Springer, Berlin Heidelberg New York, 1990.
9. B.I. Dahiyat, S.L. Mayo, De novo protein design: fully automated sequence selection, *Science* 278(5335):82-7, 1997.
10. J. Desmet, M. de Maeyer, B. Hazes, I. Lasters, The dead-end elimination theorem and its use in protein side-chain positioning, *Nature*, 356, 539-542, 1992.
11. C.A. Voigt, D.B. Gordon, S.L. Mayo, Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design, *J Mol Biol* 299(3):789-803, 2000.
12. W. Fink, unpublished information, 1998.
13. W. Fink, J.M. Dohm, M.A. Tarbell, T.M. Hare, V.R. Baker, Next-Generation Robotic Planetary Reconnaissance Missions: A Paradigm Shift, *Planetary and Space Science*, 53, 1419-1426, 2005.
14. W. Fink, J.M. Dohm, M.A. Tarbell, T.M. Hare, V.R. Baker, D. Schulze-Makuch, R. Furfaro, A.G. Fairén, T.P.A. Ferré, H. Miyamoto, G. Komatsu, W.C. Mahaney, Autonomous Tier-Scalable Reconnaissance Missions For Remote Planetary Exploration, *Proceedings of the 4th International Planetary Probe Workshop 2006, Pasadena*.
15. W. Fink, J.M. Dohm, M.A. Tarbell, T.M. Hare, V.R. Baker, D. Schulze-Makuch, R. Furfaro, A.G. Fairén, T.P.A. Ferré, H. Miyamoto, G. Komatsu, W.C. Mahaney, Tier-Scalable Reconnaissance Missions For The Autonomous Exploration Of Planetary Bodies, *IEEE Aerospace Conference Proceedings*, paper #1199, Big Sky, Montana, 2007.
16. Caltech's Visual and Autonomous Exploration Systems Research Laboratory Web site for Tier-Scalable Reconnaissance (by W. Fink) <http://autonomy.caltech.edu/autonomy/tierscalable.html>
17. Kennedy, B., Aghazarian, H., Cheng, Y., Garrett, M., Hickey, G., Huntsberger, T., Magnone, L., Mahoney, C., Meyer, A., and Knight, J., LEMUR: Legged Excursion Mechanical Utility Rover, *Autonomous Robots*, Vol. 11, No. 11, pp. 201-205, 2001.
18. W. Fink, T.L. Huntsberger, H. Aghazarian, Dynamic Optimization of N-Joint Robotic Limb Deployments, under review, 2008.
19. R.J. Terrile, C. Adami, H. Aghazarian, S.N. Chau, V.T. Dang, M.I. Ferguson, W. Fink, T.L. Huntsberger, G. Klimeck, M.A. Kordon, S. Lee, P.A. von Allmen, and J. Xu, Evolutionary Computation Technologies for Space Systems, *IEEE Aerospace Conference Proceedings*, Big Sky, MT, March 2005.
20. R.J. Terrile, H. Aghazarian, M.I. Ferguson, W. Fink, T.L. Huntsberger, D. Keymeulen, G. Klimeck, M.A. Kordon, S. Lee, and P.A. von Allmen, Evolutionary Computation Technologies for the Automatic Design of Space Systems, *NASA/DOD Evolvable Hardware Conference Proceedings*, Washington, DC, June 2005.
21. J.-C. Latombe, *Robot Motion Planning*. M.A. Norwell: Kluwer, 1991.
22. P.S. Schenker, E.T. Baumgartner, L.I. Dorsky, P.G. Backes, H. Aghazarian, J.S. Norris, T.L. Huntsberger, Y. Cheng, A. Trebi-Ollennu, M.S. Garrett, B.A. Kennedy, A.J. Ganino, R.E. Arvidson, and S.W. Squyres, FIDO: A

- Field Integrated Design & Operations rover for Mars surface exploration, *Proc. 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS-'01)*, Montreal, Canada, 2001.
23. P.S. Schenker, T.L. Huntsberger, P. Pirjanian, E.T. Baumgartner, and E. Tunstel, Planetary Rover Developments Supporting Mars Exploration, Sample Return and Future Human-Robotic Colonization, *Autonomous Robots*, No. 14, pp. 103-126, 2003.
 24. "The MEMS Handbook: MEMS Applications," ed. M. G. Hak, CRC Press, Florida, ISBN:0-8493-9139-3, 2006.
 25. T. George, "MEMS/NEMS development for Space Applications at NASA/JPL," *Proc. SPIE*, 4755, 556, 2002.
 26. T. George, "Overview of MEMS/NEMS Technology Development for Space Applications at NASA/JPL," *Proc. SPIE*, 5116, 136, 2003.
 27. M. Weinberg, J. Connelly, A. Kourepenis, and D. Sargent, "Microelectromechanical instrument and systems development at the Charles Stark Draper Laboratory, Inc.," *Proc. 16th DASC. AIAA/IEEE* 2, 33, 1997.
 28. W. Fink, T. George; M.A. Tarbell, Tier-Scalable Reconnaissance: The Challenge of Sensor Optimization, Sensor Deployment, Sensor Fusion, and Sensor Interoperability, *Proceedings of the SPIE Defense and Security Symposium*, 2007.
 29. D. Keymeulen, W. Fink, M.I. Ferguson, C. Peay, B. Oks, R. Terrile, K. Yee, Tuning of MEMS Devices using Evolutionary Computation and Open-Loop Frequency Response, *IEEE Aerospace Conference Proceedings*, Big Sky, Montana, 2005.
 30. D. Keymeulen, W. Fink, M.I. Ferguson, C. Peay, B. Oks, R. Terrile, K. Yee, Evolutionary Computation applied to the Tuning of MEMS Gyroscopes, *GECCO 2005 Conference Proceedings*, Washington DC, 2005.
 31. D. Keymeulen, M.I. Ferguson, W. Fink, B. Oks, C. Peay, R. Terrile, Y. Cheng, D. Kim, E. MacDonald, D. Floor, Hardware Platform for MEMS Gyroscopes Tuning based on Evolutionary Computation using Open-Loop and Closed-Loop Frequency Response, *ICES'2005 Conference Proceedings*, 2005.
 32. D. Keymeulen, M.I. Ferguson, W. Fink, B. Oks, C. Peay, R. Terrile, Y. Cheng, D. Kim, E. MacDonald, D. Floor, Hardware Platform for MEMS Gyroscopes Tuning based on Evolutionary Computation using Open-Loop and Closed-Loop Frequency Response, Chapter 21 in "Evolvable Systems: From Biology to Hardware", Springer Berlin/Heidelberg, 2005.
 33. D. Keymeulen, M.I. Ferguson, L. Breuer, W. Fink, B. Oks, C. Peay, R. Terrile, Y. Cheng, D. Kim, E. MacDonald, D. Floor, HARDWARE PLATFORMS FOR ELECTROSTATIC TUNING OF MEMS GYROSCOPE USING NATURE INSPIRED COMPUTATION, Chapter 10 in "Evolvable Hardware", by T. Higuchi, X.Yao, Y. Liu, Eds., Springer Verlag, 2006.
 34. S. Lee, P. von Allmen, W. Fink, A.E. Petropoulos, R.J. Terrile, Design and Optimization of Low-thrust Orbit Transfers, *IEEE Aerospace Conference Proceedings*, Big Sky, Montana, 2005.
 35. S. Lee, W. Fink, R.P. Russell, P. von Allmen, A.E. Petropoulos, R.J. Terrile, Evolutionary Computing for Low-Thrust Navigation, *Conference paper, AIAA Space Conference*, Long Beach, CA, August 30-September 1, 2005.
 36. R.J. Terrile, W. Fink, T.L. Huntsberger, S. Lee, E.R. Tisdale, G. Tinetti, P. von Allmen, Retrieval of Extra-Solar Planetary Spectra Using Evolutionary Computational Methods, Division for Planetary Sciences (DPS) 37th Meeting of the American Astronomical Society, Cambridge, UK, *Bull. Amer. Astron. Soc.*, 37, 31.19, 2005.
 37. R.J. Terrile, S. Lee, G. Tinetti, W. Fink, E.R. Tisdale, T.L. Huntsberger, P. von Allmen, Spectral Retrieval of Exoplanetary Signatures Using Evolutionary Computational Methods, Proceeding of the EGU General Assembly, Vienna, Austria, 2006.
 38. R.J. Terrile, G. Tinetti, S. Lee, W. Fink, T.L. Huntsberger, P. von Allmen, E.R. Tisdale, Retrieval of Earthshine Spectra Using Evolutionary Computational Methods as Analogs for Extra-Solar Planetary Spectra, American Geophysical Union, Baltimore, MD, 2006.
 39. R.J. Terrile, S. Lee, G. Tinetti, W. Fink, T.L. Huntsberger, P. von Allmen, Evolutionary Computational Methods for the Design of Spectral Instruments, *IEEE Aerospace Conference Proceedings*, paper #1194, Big Sky, Montana, 2008.
 40. N.J. Woolf, P.S. Smith, W.A. Traub, and K.W. Jucks, The spectrum of earthshine: A pale blue dot observed from the ground, *Astrophysical Journal*, 574 (1), 430-433, 2002.
 41. W. Fink, Phasenraumanalyse von Modellen neuronaler Netzwerke (Coupling space analysis in models of neural networks), Diplomarbeit (Master's Thesis), Institute for Theoretical Physics, University of Göttingen, 1993.
 42. Official Huygens Search & Optimisation Benchmarking Suite Web site <http://gungurru.csse.uwa.edu.au/cara/huygens/>
 43. Caltech's Visual and Autonomous Exploration Systems Research Laboratory Web site for "Huygens Probe" Optimization Competition (by W. Fink) http://autonomy.caltech.edu/evolutionary_computation/CEC2006.html